



*Mój przyjacielu, byłeś mi naprawdę bliski,
mój przyjacielu, wiesz, że byłeś mi, jak brat?
Dałem ci kabel, dałem ci pokój,
dałem bitarrę, bitcoinów dochód
i system Linux, a na Windowsa wszedłeś sam.*

Bit Jovi to nowy, aspirujący artysta w królestwie Bajtocji. Jego głos jest prawdziwie nadzwyczajny, a przynajmniej tak mówiła mu jego mama. Niestety, czymże jest sam wokal bez żadnych instrumentów? Bit nie jest fanem stylu *a capella*, a nie stać go jeszcze na zatrudnienie całego zespołu. Postanowił więc wziąć sprawy w swoje ręce. Na strychu udało mu się wygrzebać starą bitarrę klasyczną swojego ojca. Zaopatrzył się też w profesjonalny poradnik „Bitarra. Podstawy gry z płytą CD”, którą znalazł w lokalnym sklepie spożywczym. Po kilku dniach lektury Bit był już pewny – żeby zagrać swoje ulubione utwory, takie jak „Never gonna bit you up”, „Sweet Keyboard O’Mine” czy „E-mail do M.”, koniecznie musi nauczyć się akordów.

Oto fragment z jego profesjonalnej książki:

„W muzyce zachodniej rozróżniamy dwanaście dźwięków: A, A#, B, C, C#, D, D#, E, F, F#, G i G#. Jak łatwo zauważyć, można je reprezentować jako kolejne liczby naturalne od 0 do 11. Odległość między dwoma sąsiednimi dźwiękami (czyli dźwiękami, którym przyporządkowane wartości liczbowe różnią się o 1) nazywamy półtonem. Dźwięk wyższy o pół tonu od A to A#, natomiast od G# to A – mamy tutaj do czynienia z dodawaniem modulo 12.

Akord durowy to akord składający się z dźwięku bazowego (prymy), dźwięku o 4 półtony wyższego od prymy (tercji wielkiej) oraz dźwięku o 7 półtonów wyższego od prymy (kwinty czystej). Na przykład, akord E-dur składa się z prymy E, tercji wielkiej G# oraz kwinty czystej B (E możemy reprezentować liczbą 7, wtedy w akordzie będziemy mieć dźwięki reprezentowane przez liczby $7 + 4 = 11$ oraz $(7 + 7) \bmod 12 = 2$).

Podobnie, akord molowy składa się z prymy, dźwięku o 3 półtony wyższego od prymy (tercji małej) oraz kwinty czystej (dźwięku o 7 półtonów wyższego od prymy). Na przykład, akord A-moll składa się z dźwięków A, C oraz E.”

Należy zauważyć, że jeden zbiór trzech dźwięków może reprezentować co najwyżej jeden z powyższych akordów.

„Bitarra. Podstawy gry z płytą CD”

Prawdę mówiąc, Bit nic z tego nie rozumie. Cały ten opis wydaje mu się jakimś matematycznym bełkotem! Co gorsze, pierwszy koncert solowy zaplanowany jest już na przyszły miesiąc! Pomóż Bitowi rozszyfrować, czy i jaki akord tworzą interesujące go trzy nuty.

WEJŚCIE

W jedynym wierszu wejścia znajdują się trzy ciągi znaków S_1, S_2, S_3 oddzielone pojedynczą spacją, oznaczające trzy nuty. Ciąg reprezentujący nutę złożony jest z jednej dużej litery alfabetu angielskiego (od A do G) lub z dużej litery alfabetu angielskiego i następującego po niej znaku #.

WYJŚCIE

Jeżeli podane na wejściu nuty tworzą akord, program powinien wypisać słowo Akord oraz oddzielony spacją ciąg znaków reprezentujący prymę (dźwięk bazowy akordu, opis ciągu reprezentującego daną nutę znajduje się w specyfikacji wejścia), a w przypadku akordu molowego, z dopisaną na końcu literą m. W przeciwnym przypadku program powinien wypisać Brak.

PRZYKŁAD

Wejście

E A B

Wejście

E G# B

Wejście

C D# G

Wejście

F# A# D#

Wyjście

Brak

Wyjście

Akord E

Wyjście

Akord Cm

Wyjście

Akord D#m

Zagadka

Zadanie: **BBB**
Limit pamięci: **256 MB**
Limit czasu: **1 s**



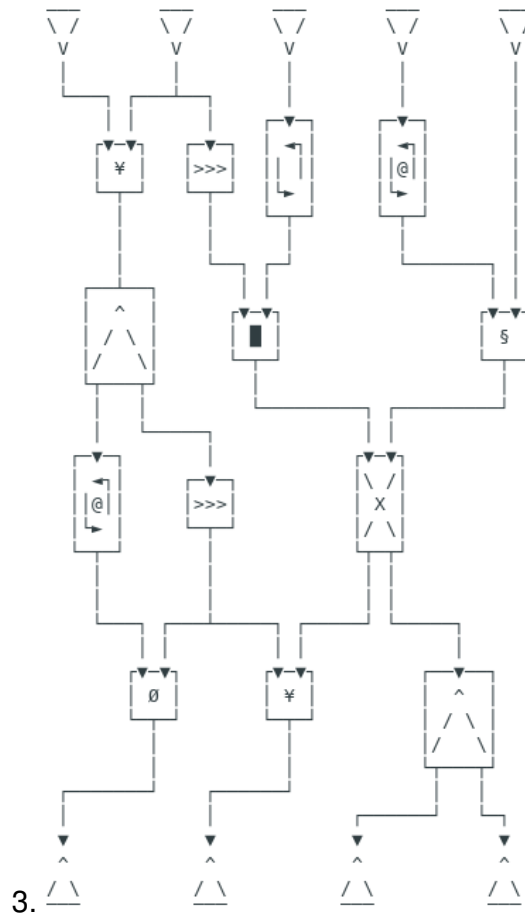
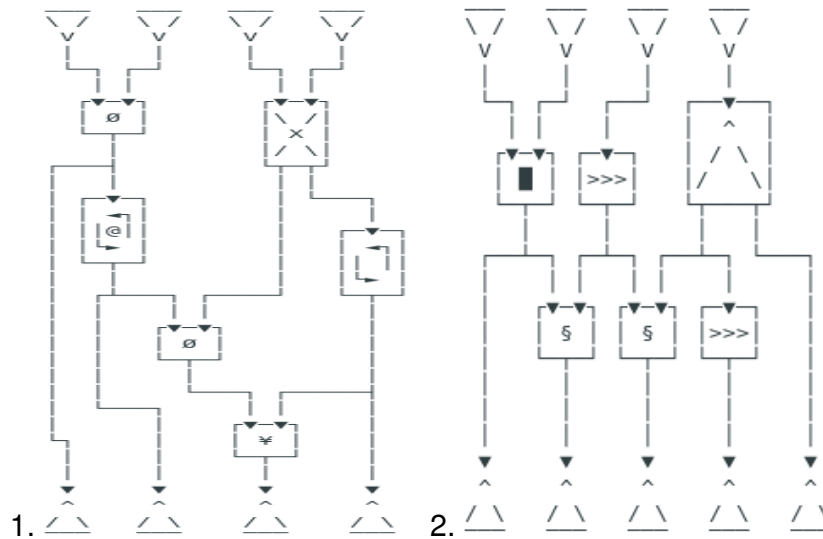
Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Nie ma programistów? A dlaczego tu nie ma okienek? A dlaczego tu nie ma myszki?

Janusz, reprezentant prawie wymarłego już zawodu programisty, otrzymał do rozwiązania najważniejsze zadanie, które zwińczy jego wspaniałą karierę. Dostał trzy schematy zapisane w starożytnym języku ascików, dwa wysłużone komputery w kształcie komody o wysokości 64 cm oraz jedno urządzenie wejścia w postaci klawiatury typu M. Schematy, które otrzymał wyglądają następująco:



Po kilku chwilach odkrył, że komputery mają zaprogramowane dwa pierwsze schematy, trójkąty na górze schematów symbolizują wejścia, a te na dole – wyjścia.

Kolejne minuty zmagani przyniosły kolejne wnioski: zarówno pojedyncze wejście, jak i wyjście ma formę kwadratu złożonego z 16 starożytnych symboli: + lub #.

Pozostało mu jedynie wymyślić, co skrywają schematy i stworzyć program, który swoim działaniem będzie symulował ostatni z nich. Januszowi udało się uporać ze wszystkim w pięć godzin.

Czy Ty będziesz szybszy? **Tutaj** znajdziesz dwa interfejsy, podobne do tych, z których korzystał też Janusz, a które **powinny Ci pomóc w rozwiązaniu zadania**. Dla Twojej wygody dodane zostało GUI oraz obsługa myszki ;)

WEJŚCIE

Kolejne wejścia do programu oddzielone są pojedynczym pustym wierszem. Każde wejście składa się z czterech wierszy, a każdy wiersz zawiera cztery znaki, które mogą być równe + lub #.

WYJŚCIE

Format wyjścia jest identyczny z formatem wejścia. Twój program powinien wypisać wyjścia oddzielone pojedynczym pustym wierszem. Każde wyjście powinno składać się z czterech wierszy, a każdy wiersz powinien zawierać cztery znaki równe + lub #.

PRZYKŁAD

Wejście

```
+###  
###  
+###  
####
```

Wyjście

```
++++  
++++  
++##  
+###
```

```
+###  
+###  
++##  
###
```

```
++++  
++++  
++##  
++++
```

```
+###  
++##  
####  
####
```

```
++##  
++++  
####  
++++
```

```
+++##  
+###  
##+##  
##+##
```

```
++++  
++##  
++++  
++++
```

```
##+##  
##+##  
++++  
####
```

Sorry, taki mamy klimat

Zadanie: **CCC**
Limit pamięci: **256 MB**
Limit czasu: **1 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Sorry, taki mamy klimat.

Po latach pracy w korporacji Janusz przeszedł na emeryturę i kupił działkę za miastem, na której postanowił hodować bluszcz. Planuje wybudować podpórki, aby jego ulubione rośliny miały się po czym piąć. Na osi północ-południe w odstępach jednostkowych wbił N pionowych pali o parami różnych wysokościach.

Przygotował też N poprzeczek o całkowitych długościach. Chciałby je przyczepić do szczytów pali w taki sposób, żeby:

- do każdego pala była przymocowana co najwyżej jedna poprzeczka,
- każda poprzeczka znajdowała się idealnie w linii północ-południe i wisiała poziomo,
- każda poprzeczka dotykała tylko tego pala, do którego jest przymocowana.

Aby udało mu się zamontować wszystkie poprzeczki zgodnie z powyższymi warunkami, niektóre z nich trzeba by było skrócić. Jednak piła mechaniczna, którą Janusz trzymał przez zimę w szopie, nie wytrzymała tegorocznych trzaskających mrozów, więc przycinanie poprzeczek nie wchodzi w grę. Być może jednak jest jeszcze nadzieja? Oblicz maksymalną liczbę poprzeczek, które Janusz może przymocować do pali.

WEJŚCIE

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N oznaczająca liczbę pali i poprzeczek. W drugim wierszu jest N liczb naturalnych d_1, d_2, \dots, d_N oddzielonych pojedynczym odstępem, które oznaczają długości poprzeczek. Trzeci wiersz składa się z N parami różnych liczb naturalnych h_1, h_2, \dots, h_N oznaczających wysokości kolejnych pali.

WYJŚCIE

Twój program w jedynym wierszu wyjścia powinien wypisać jedną nieujemną liczbę całkowitą – maksymalną liczbę poprzeczek, które można przyczepić do pali zgodnie z zasadami.

OGRANICZENIA

$1 \leq N \leq 500\,000, 1 \leq d_i, h_i \leq N.$

PRZYKŁAD

Wejście

4
2 3 4 4
3 1 2 4

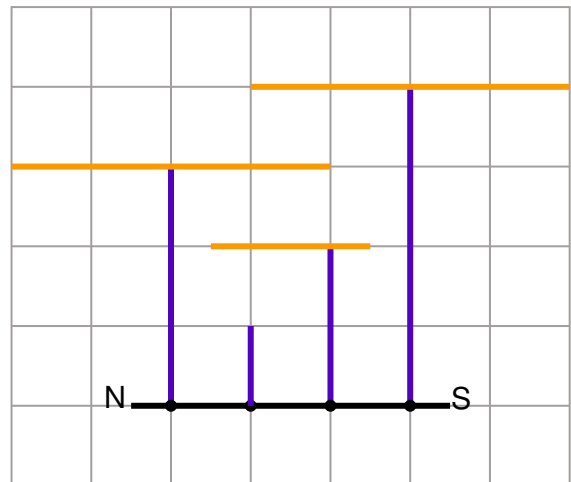
Wyjście

3

Do drugiego pala od lewej można przymocować tylko poprzeczki krótsze niż 2, więc nie da się przymocować wszystkich poprzeczek. Natomiast można przymocować trzy z nich, na przykład w taki sposób:

- Do pala o wysokości 3 przymocować poprzeczkę długości 4 na jej środku.
- Do pala o wysokości 2 przymocować poprzeczkę długości 2 w odległości $\frac{1}{2}$ od jej prawego końca.
- Do pala o wysokości 4 przymocować poprzeczkę długości 4 na jej środku.

Poniższy rysunek przedstawia to rozwiązanie:



Wejście

4
4 4 4 4
1 2 3 4

Wyjście

4

Płot

Zadanie: **DDD**
Limit pamięci: **256 MB**
Limit czasu: **1 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Mur naprawia, mur graniczny, trzech mularzy, on rozkazał! On się waży!...

Nadszedł weekend i Janusz rozpoczął swój relaks po tygodniu ciężkiej pracy. Niestety, już po kilku minutach spokoju usłyszał krzyk dzieci sąsiada, a po chwili także ujadanie ich psa. Okazało się, że ich ulubiona piłka wpadła na działkę Janusza. Chcąc nie chcąc wstał on z ulubionego fotela i odrzucił dzieciom zabawkę. Aby uchronić się przed kolejnymi takimi sytuacjami i zaznać upragnionego odpoczynku, postanowił naprawić płot. Po zmierzeniu swojej działki Janusz obliczył, że potrzebuje K desek.

Szybko udał się do pobliskiego sklepu i kupił potrzebne materiały. Niestety okazało się, że niektóre deski uszkodziły się w trakcie transportu i Januszowi pozostało N z nich. Na domiar złego sztachety nie są równej długości, więc musi je pociąć w taki sposób, żeby mieć K równych fragmentów.

Zastanawia się teraz czy w ogóle jest sens zaczynać budowę. W końcu, na co zdałyby się wszystkie jego wysiłki, gdyby płot w naprawionym miejscu był za niski? Dlatego poprosił Cię o napisanie programu, który obliczy maksymalną długość desek, jaką może otrzymać. Jako że wynik może być liczbą wymierną powinieneś wypisać go w odpowiednim formacie, o którym mowa poniżej.

WEJŚCIE

Pierwszy wiersz zawiera dwie liczby N oraz K . Drugi wiersz zawiera N liczb naturalnych A_1, A_2, \dots, A_N podzielanych pojedynczym odstępem. Liczba A_i oznacza długość i -tej deski.

WYJŚCIE

Twój program powinien wypisać dwie względnie pierwsze liczby naturalne P, Q , oznaczające, że wynik wynosi $\frac{P}{Q}$.

OGRANICZENIA

$1 \leq N \leq 100\,000$, $1 \leq K \leq 10^9$, $1 \leq A_i \leq 10^9$.

PRZYKŁAD

Wejście

2 3
5 2

Wyjście

2 1

Znowu ten płot

Zadanie: **EEE**
Limit pamięci: **256 MB**
Limit czasu: **1 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Kargul, podejdź no do płota!

Tym razem zostałeś poproszony o pomoc w naprawie nie byle jakiego płotu. Nasi bohaterowie wspólnymi siłami starają się go wyrównać, to znaczy chcą zmaksymalizować liczbę sztachet mających taką samą wysokość. W tym celu jeden z nich może zmniejszyć, a drugi zwiększyć wysokości pewnej liczby desek.

Niestety nasi przyjaciele nie są zbyt zdecydowani co do tego, jakie sztachety ostatecznie powinien zawierać płot, dlatego co rusz dokładają i wyciągają je z niego. Czy jesteś w stanie im pomóc?

Twój program powinien obsługiwać następujące rodzaje operacji na (początkowo pustym) płocie:

- $D X Y$ - Wstaw Y sztachet o wysokości X do płotu.
- $U X Y$ - Wyciągnij Y sztachet o wysokości X z płotu (możesz założyć, że jest ich tam co najmniej tyle).
- $Z A B$ - Oblicz maksymalną liczbę sztachet mogących mieć taką samą wysokość, jeśli nasi bohaterowie mogą zwiększyć wysokości co najwyżej A z nich oraz zmniejszyć wysokości co najwyżej B z nich.

WEJŚCIE

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą Q będącą liczbą operacji, które powinien wykonać Twój program. Każdy z następujących Q wierszy zawiera dokładnie jedną operację podaną w formacie opisanym powyżej.

WYJŚCIE

Dla każdej operacji typu Z powinieneś wypisać jeden wiersz zawierający jedną liczbę całkowitą – maksymalną liczbę sztachet mających taką samą wysokość, zgodnie z warunkami podanymi powyżej. Możesz założyć, że na wejściu zawsze będzie co najmniej jedna operacja tego typu.

OGRANICZENIA

$1 \leq Q \leq 300\,000$, $1 \leq X, Y \leq 10^9$, $0 \leq A, B \leq 10^{18}$.

PRZYKŁAD

Wejście	Wyjście
5	4
D 1 3	3
D 2 2	
Z 1 1	
U 1 2	
Z 1 0	

Zamarznięte zwrotnice

Zadanie: **FFF**
Limit pamięci: **256 MB**
Limit czasu: **1 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Ale urwał! Ale to było dobre!

W Bajtocji nastała sroga zima. Samochody ślizgają się po ulicach, a zamrożone zwrotnice blokują tramwaje w całym królestwie. Hrabia Biteusz Zwrotniarz, doktor habilitowany nauk tramwajowych, został zawezwany na dwór królewski, aby rozwiązać problem z transportem w stolicy.

Niestety, nie uda się w krótkim czasie naprawić mechanizmu zwrotnic. Hrabia zaproponował więc, żeby ustawić wszystkie zwrotnice w konkretnej pozycji. Oczywiście należy to zrobić tak, aby jak najwięcej mieszkańców mogło korzystać ze swojej ulubionej linii tramwajowej.

W stolicy znajduje się N zajezdni tramwajowych ponumerowanych kolejnymi liczbami naturalnymi. Niektóre zajezdnie połączone są ze sobą torami tak, że pomiędzy dowolną parą zajezdni można przejechać tramwajem na dokładnie jeden sposób. Ze względu na zamrożone zwrotnice, Hrabia musi dla każdej zajezdni wybrać dokładnie dwa tory, między którymi będzie można przejechać (w obie strony), lub jeden tor, na którym będzie można zawrócić.

Dotychczas w stolicy było Q linii tramwajowych ponumerowanych kolejnymi liczbami naturalnymi. Linia o numerze i rozpoczynała się w zajezdni a_i i kończyła w zajezdni b_i , a regularnie korzystało z niej w_i mieszkańców. Po ustawieniu przez Hrabiego zwrotnic dana linia będzie dalej funkcjonować, jeżeli będzie dało się przejechać tramwajem od początkowej do końcowej zajezdni, a na torach połączonych bezpośrednio z tymi dwoma zajezdniami będzie można zawrócić.

Pomóż Hrabiemu ustawić zwrotnice tak, aby suma mieszkańców regularnie korzystających z funkcjonujących linii tramwajowych była jak największa.

WEJŚCIE

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N oznaczająca liczbę zajezdni. W każdej z kolejnych $N - 1$ linii znajdują się dwie liczby całkowite c_i i d_i oznaczające, że zajezdnie c_i i d_i są połączone torem.

W kolejnym wierszu wejścia znajduje się liczba Q oznaczająca liczbę linii tramwajowych, które funkcjonowały przed nadejściem zimy. W następnych Q wierszach znajdują się trzy liczby całkowite a_i , b_i , w_i – numer początkowej i końcowej zajezdni i -tej linii tramwajowej i liczba regularnie korzystającej z niej mieszkańców.

WYJŚCIE

W jedynej linii wyjścia program powinien wypisać jedną liczbę całkowitą - największa możliwa suma mieszkańców regularnie korzystających z funkcjonujących linii tramwajowych po ustawieniu zwrotnic przez Hrabiego.

OGRANICZENIA

$2 \leq N \leq 100\,000$, $1 \leq Q \leq 100\,000$, $1 \leq a_i, b_i, c_i, d_i \leq N$, $1 \leq w_i \leq 10^9$.

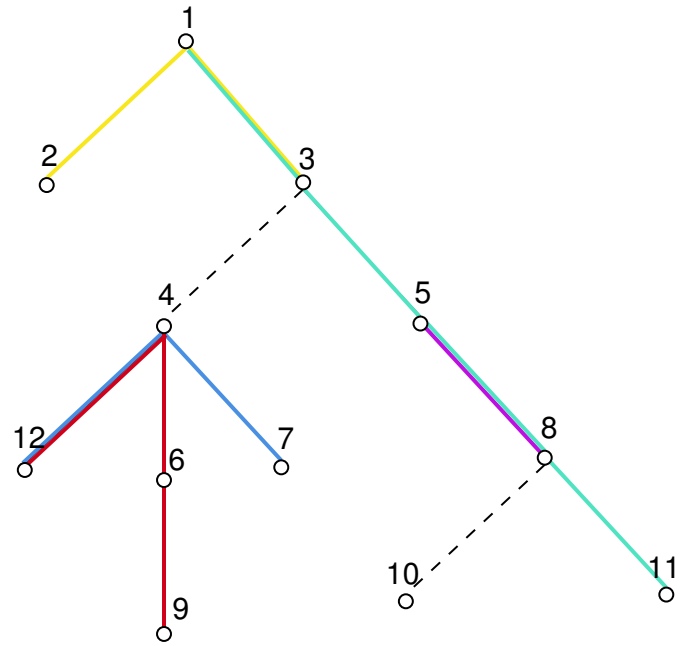
PRZYKŁAD

Wejście

12
1 2
1 3
3 4
3 5
4 12
4 6
4 7
6 9
5 8
8 10
8 11
5
1 11 1
5 8 1
2 3 1
12 7 1
12 9 1

Wyjście

3



Truskawka na torcie

Zadanie: **GGG**
Limit pamięci: **256 MB**
Limit czasu: **5 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Asysta Zielińskiego to taka truskawka na torcie.

Jasio poświęcił dziś cały dzień na naukę algorytmiki. Po wielogodzinnych, ale i owocnych zmaganiach z kombinatoryką oraz teorią liczb doszedł do ostatniego zadania, które jest prawdziwą truskawką na torcie, bo łączy w sobie ulubione przez Jasia zagadnienia: drzewa i algorytmy tekstowe.

W zadaniu dane jest drzewo o N wierzchołkach etykietowanych liczbami naturalnymi od 1 do N ukorzenione w wierzchołku 1. Na każdej z jego krawędzi napisana jest jedna mała litera angielskiego alfabetu. Wystąpieniem słowa s nazwiemy ścieżkę z pewnego wierzchołka v w górę drzewa (w stronę korzenia) taką, że litery na kolejnych krawędziach układają się w słowo s . Mówimy, że takie wystąpienie zaczyna się w wierzchołku v .

Dane jest Q zapytań, i -te z nich dotyczy słowa s_i oraz wierzchołka v_i . Należy obliczyć liczbę wierzchołków w poddrzewie v_i , w których zaczyna się w nich pewne wystąpienie s_i . Aby ograniczyć rozmiar wejścia, słowo s_i jest opisane przez jego długość oraz pewien wierzchołek (niekoniecznie z poddrzewa v_i), w którym zaczyna się wystąpienie s_i .

Zadanie polega na tym, żeby odpowiedzieć na wszystkie zapytania poprawnie i szybko. Jasio umie niestety tylko albo jedno, albo drugie. Nie byłoby w tym nic złego, gdyby nie to, że nie dostanie deseru, jeśli nie zrobi wszystkich zadań zaplanowanych na dziś. Jako że to zadanie zostało mu jako ostatnie, Jasio poprosił Cię o pomoc. Rozwiąż zadanie za niego, a może podzieli się z Tobą nagrodą. Warto, bo dzisiaj jego mama zrobiła tort truskawkowy.

WEJŚCIE

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne N i Q oznaczające odpowiednio rozmiar drzewa i liczbę zapytań.

Następne $N - 1$ wierszy opisuje krawędzie, i -ty z nich składa się z dwóch liczb naturalnych i jednej małej litery angielskiego alfabetu: a_i, b_i, c_i . Oznacza to, że wierzchołek a_i jest rodzicem wierzchołka b_i , a na krawędzi łączącej wierzchołek a_i z b_i jest napisana litera c_i .

Kolejne Q wierszy opisuje zapytania. W i -tym znajdują się trzy liczby: v_i, t_i oraz l_i , które oznaczają, że słowo s_i ma długość l_i i pewne jego wystąpienie zaczyna się w t_i .

WYJŚCIE

Twój program powinien wypisać na standardowe wyjście Q wierszy, i -ty z nich powinien zawierać jedną liczbę całkowitą – liczbę wystąpień słowa s_i zaczynających się w poddrzewie wierzchołka v_i .

OGRANICZENIA

$2 \leq N \leq 100\,000$, $1 \leq Q \leq 100\,000$, $1 \leq a_i, b_i, v_i, t_i \leq N$, $1 \leq l_i < N$.

PRZYKŁAD

Wejście

6 3
1 2 a
2 3 a
1 4 a
2 5 b
3 6 b
2 4 1
1 5 2
3 5 2

Wyjście

2
2
1

Pierwsze zapytanie dotyczy słowa a, którego wystąpienia zaczynają się w wierzchołkach 2, 3 i 4. Kolejne dwa zapytania są o słowo ba. Jego wystąpienia zaczynają się w wierzchołkach 6 i 5.

Wejście

8 4
1 2 c
2 6 a
1 3 a
3 4 b
3 5 b
5 7 a
7 8 b
1 6 1
3 6 2
3 4 2
7 8 4

Wyjście

3
0
3
1

Malowanie

Zadanie: **HHH**
Limit pamięci: **256 MB**
Limit czasu: **1 s**



Ministerstwo
Cyfryzacji



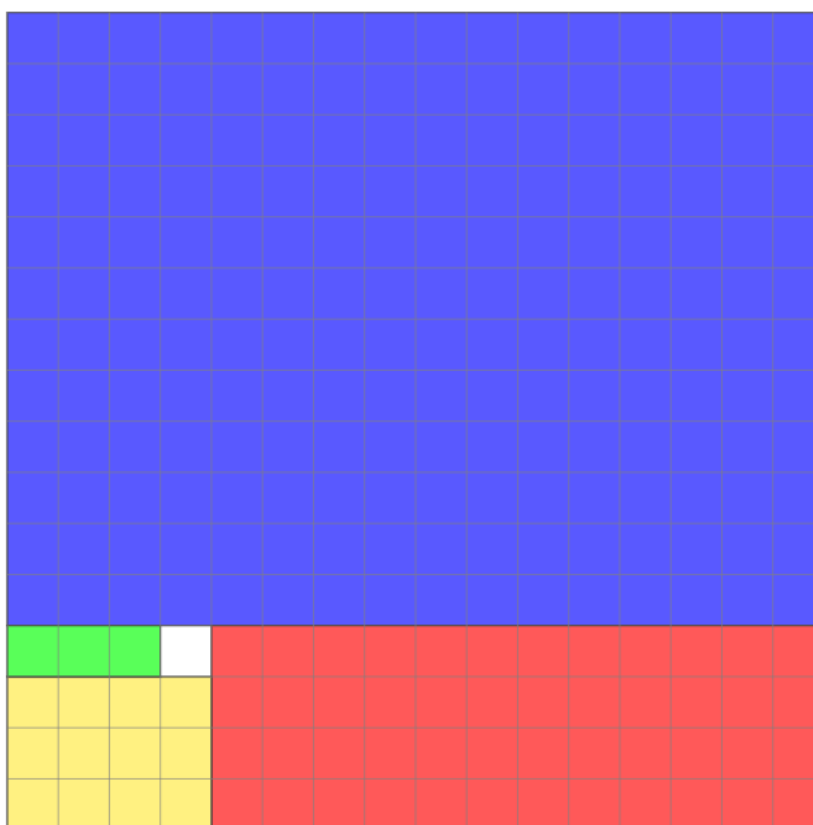
Uniwersytet
Wrocławski

To je amelinium, tego nie pomalujesz.

Przyszła wiosna, a wraz z nią nadszedł czas na remont. Pomimo że Jasio jest wybitnym informatykiem i nie lubi pracy fizycznej, jego tata zlecił mu ważne zadanie. Młody programista musi pomalować ścianę w swoim pokoju, która ma W milimetrów wysokości i S milimetrów szerokości. Na domiar złego, w odległości y milimetrów od podłogi i x milimetrów od lewej krawędzi znajduje się kwadratowy gwóźdź zasłaniający fragment ściany o wymiarze 1×1 milimetr, więc Jasiu nie będzie mógł pomalować tego kawałka.

Aby uprzyjemnić sobie pracę i rozwinąć swoją informatyczną pasję, wymyślił on prosty algorytm, który pomoże mu wykonać zadanie. Cały proces będzie podzielony na fazy, pomiędzy którymi Jasio będzie robił przerwy na programowanie. Na początku każdej z faz wybierze on jedną z krawędzi niepomalowanego dotąd obszaru, od której zacznie malowanie. Długimi pociągnięciami pędzla będzie nakładał kolejne warstwy farby jedna obok drugiej, aż pomaluje 75% obszaru. Tym samym jeden z wymiarów pozostałego fragmentu do pomalowania zmaleje czterokrotnie, a drugi pozostanie bez zmian.

Poniższy rysunek ilustruje możliwe malowanie w sytuacji opisanej w trzecim przypadku testowym:



Kolorem niebieskim zaznaczony jest obszar pomalowany przez Jasia w pierwszej fazie, czerwonym w drugiej, żółtym w trzeciej i zielonym w czwartej. Na białym polu znajduje się gwóźdź.

Jasio zastanawia się, czy może wybrać taką kolejność działań, żeby pomalować całą ścianę, ale nie pomalować gwoźdźcia.

WEJŚCIE

Pierwszy wiersz wejścia zawiera jedną liczbę T oznaczającą liczbę przypadków testowych. Następnie znajduje się T wierszy. i -ty wiersz zawiera cztery liczby W, S, Y, X oznaczające odpowiednio wysokość i szerokość ściany oraz położenie gwoźdźcia w i -tym przypadku testowym.

WYJŚCIE

Twój program powinien wypisać i wierszy. W i -tym wierszu powinna się znaleźć odpowiedź na i -ty przypadek testowy:

- Tak, jeżeli istnieje poprawna kolejność malowania w i -tym przypadku testowym,
- Nie – w przeciwnym przypadku.

OGRANICZENIA

$1 \leq T \leq 100\,000$, $1 \leq W, S \leq 10^{18}$, $0 \leq X < S$, $0 \leq Y < W$.

PRZYKŁAD

Wejście	Wyjście
3	Nie
3 4 0 0	Nie
4 4 2 2	Tak
16 16 3 3	

Mafia

Zadanie: III
Limit pamięci: 256 MB
Limit czasu: 2 s



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Przychodzisz do mnie w dniu konkursu i prosisz bym rozwiązywał problemy za akcepty.

Dzięki wieloletniemu przemykaniu oka przez władze przestępczość zorganizowana stała się sporym problemem w Bajtocji. Co prawda działalność lokalnej mafii ogranicza się do handlu nielegalnymi kopiami Windowsa, ale nowo wybrany komendant bajtockiej policji postanowił zająć się w końcu tą sprawą.

Miasto Bajtocja podzielone jest na dzielnice, kontrolowane przez zwaśnione rodziny przestępcze. Co ciekawe, jeżeli przedstawimy dzielnice jako wierzchołki, a między sąsiadującymi dzielnicami narysujemy krawędzie, to otrzymamy drzewo, czyli spójny graf bez cykli. Komendant wpadł na chytry plan, żeby wysłać szpiegów (tak zwane wtyki) do każdej z dzielnic. Niestety dostępnych tajniaków nie jest wielu, więc niektórzy z nich będą musieli szpiegować w kilku rodzinach jednocześnie. W trosce o bezpieczeństwo szpiegów, dowolne dwie dzielnice, do których zostanie wysłany agent, muszą być oddalone od siebie o co najmniej dwie inne dzielnice.

Twoim zadaniem jest napisanie programu, który ustali, ilu szpiegów potrzeba do zrealizowania planu komendanta oraz podanie ich przykładowego rozmieszczenia.

Komendant chciałby też zobaczyć, jak wyglądałoby rozmieszczenie w innych, podobnych do Bajtocji miastach. Podaj odpowiedź dla T niezależnych przypadków.

WEJŚCIE

W pierwszym wierszu wejścia znajduje się T – liczba niezależnych przypadków do rozwiązania. Opis każdego przypadku zaczyna się od wiersza z liczbą N , oznaczającą liczbę dzielnic. W każdym z następnych $N - 1$ wierszy znajdują się dwie liczby całkowite oddzielone pojedynczym odstępem, oznaczające numery sąsiadujących dzielnic.

WYJŚCIE

Dla każdego przypadku należy wypisać dwa wiersze. W pierwszym z nich ma się znajdować liczba całkowita K , oznaczająca liczbę potrzebnych szpiegów. W drugim należy podać ciąg N liczb całkowitych z przedziału od 1 do K włącznie – numery szpiegów przypisanych do kolejnych dzielnic.

OGRANICZENIA

$1 \leq N, T \leq 500\,000, \sum N \leq 500\,000$.

PRZYKŁAD

Wejście	Wyjście
3	3
5	1 3 2 1 3
1 2	4
2 3	1 2 3 4
3 4	1
4 5	1
4	
1 2	
1 3	
1 4	
1	

Grzegorz i huśtawka

Zadanie: **JJJ**
Limit pamięci: **256 MB**
Limit czasu: **1 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

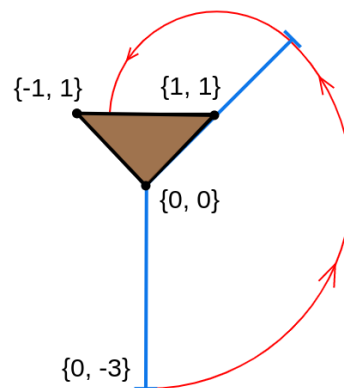
Czy często się Pan uśmiecha?

Grzegorz jest szkolnym łobuzem. Co prawda marzy o tym, żeby w przyszłości zostać doktorem nauk, ale na chwilę obecną zapytany o to co z czego się śmieje najczęściej, odpowiedziałby zapewne, że z tego, jak dzieci nie mogą się bawić na placu zabaw.

Grzegorz odwiedził ostatnio pobliski plac zabaw i postanowił na nim zablokować huśtawkę, przez okręcenie jej wokół belki, na której jest zawieszona. Ponieważ lubi trenować swoje umiejętności matematyczne, nie omieszkał policzyć, jak długą drogę przebędzie siodelko huśtawki, zanim uderzy belkę w trakcie obracania się wokół niej.

Przekrój belki jest wielokątem wypukłym o N wierzchołkach, który możemy reprezentować na dwuwymiarowej płaszczyźnie za pomocą zbioru punktów o współrzędnych całkowitych. Pierwszy wierzchołek zawsze ma współrzędne $\{0, 0\}$. Kolejne wierzchołki są podane w kolejności odwrotnej do kierunku wskazówek zegara. Każdy z nich ma **dodatnią drugą współrzędną**. Zwisająca swobodnie linka huśtawki o długości D i pomijalnie małej grubości jest przymocowana do pierwszego wierzchołka, a na jej końcu znajduje się siodelko.

Zakładając, że Grzegorz popchnie siodelko wystarczająco szybko, poziomo, w kierunku przeciwnym do kierunku wskazówek zegara, napisz program, który obliczy drogę przebytą przez siodelko, zanim uderzy ono w belkę.



WEJŚCIE

W pierwszym wierszu standardowego wejścia znajdują się dwie dodatnie liczby całkowite N oraz D oddzielone pojedynczym odstępem oznaczające odpowiednio liczbę wierzchołków reprezentujących belkę oraz długość linki huśtawki. Następne N linii zawiera opis punktów składający się z dwóch liczb całkowitych x_i, y_i oddzielonych pojedynczym odstępem, oznaczających współrzędne i -tego wierzchołka w przekroju belki. Możesz założyć, że punkty tworzą wielokąt wypukły oraz są podane w kolejności odwrotnej do kierunku wskazówek zegara. Co więcej, przyjmij że $x_1 = y_1 = 0$ oraz że żadne trzy punkty na wejściu nie są współliniowe.

WYJŚCIE

W pierwszym (jedynym) wierszu standardowego wyjścia należy umieścić jedną liczbę rzeczywistą - sumaryczną drogę siodelka huśtawki, zanim uderzy ona w belkę. Twoja odpowiedź zostanie uznana za prawidłową, jeżeli błąd względny lub bezwzględny wyniku nie przekroczy wartości 10^{-6} .

OGRANICZENIA

$3 \leq N \leq 100\,000$, $1 \leq D \leq 10^9$, $-10^9 \leq x_i \leq 10^9$, $0 \leq y_i \leq 10^9$.

PRZYKŁAD

Wejście

3 3
0 0
1 1
-1 1

Wyjście

10.8050047375

Wejście

4 30
0 0
1 3
-1 2
-1 1

Wyjście

409.2057924488



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Szyny były złe.

Bajtocja jest krajem składającym się z N miast ułożonych na linii prostej, dzięki czemu ich położenie możemy reprezentować jako współrzędne x na osi liczbowej. W każdym z miast znajduje się stacja kolejowa.

Z racji, że ostatnimi czasy król Jan dostawał bardzo dużo doniesień na temat tego, że szyny są złe i pociągi się często wykoleją, postanowił wreszcie wziąć sprawę w swoje ręce i wybudować zupełnie nową sieć linii kolejowych. Teraz jego celem jest zaplanowanie jak mają przebiegać tory.

Król ma możliwość wybudowania torowiska z dowolnego miasta do dowolnego innego, tak aby pomijały one miasta pośrednie. Długość takiego torowiska jest równa różnicy współrzędnych x tych dwóch miast. Sieć oczywiście musi być tak zaplanowana, żeby z każdego miasta dało się dojechać do dowolnego innego. Jednakże, ze względu na to, że mieszkańcy nie lubią przejeżdżać przez inne stacje podczas swojej podróży, król wydał rozporządzenie, że *ze stolicy Bajtocji do każdego innego miasta musi istnieć trasa, która przejeżdża przez co najwyżej K stacji, włącznie ze stacją początkową i końcową.*

Teraz król zastanawia się jaka jest minimalna długość torowisk potrzebna do wybudowania linii kolejowej, spełniającej wszystkie powyższe warunki. Czy jesteś w stanie mu pomóc?

WEJŚCIE

Pierwszy wiersz wejścia zawiera trzy liczby całkowite N , K i S , oznaczające liczbę miast w Bajtocji, maksymalną liczbę stacji na trasach ze stolicy do innych miast oraz numer miasta, które jest stolicą. Drugi wiersz wejścia zawiera liczby a_1, a_2, \dots, a_N , oznaczające współrzędne kolejnych miast na mapie (prostej).

WYJŚCIE

Na wyjściu wypisz jedną liczbę – minimalny koszt budowy sieci kolejowej spełniającej warunki z zadania. Jeżeli takiej sieci nie da się stworzyć, wypisz na wyjściu -1 .

OGRANICZENIA

$1 \leq K, S \leq N \leq 200, K \leq 50, 1 \leq a_1 < a_2 < \dots < a_N \leq 10^9$.

PRZYKŁAD

Wejście

7 3 4
1 2 3 4 5 6 7

Wyjście

8

Wejście

8 8 8
1 2 13 24 55 86 107 108

Wyjście

107

Liczby Ryby

Zadanie: **LLL**
Limit pamięci: **256 MB**
Limit czasu: **0.5 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Ty długo byłeś dobry, Kiler, ale wpadłeś. Bo ja byłem lepszy. Jakieś czterdzieści sześć razy lepszy. I dlatego do końca życia będziesz oglądał świat w kratkę. O!

Jerzemu Rybie nie było łatwo wyliczyć, ile razy jest lepszy od Kilera. Te wyliczenia okazały się być na tyle skomplikowane, że potrzebuje Twojej pomocy.

Przez N oznaczmy liczbę nieudanych pogoni Ryby za Kilerem. Ryba przez A , B i C oznaczył współczynniki pogoni, które są całkowitymi liczbami dodatnimi nie większymi niż N oraz które spełniają następującą zależność:

$$N = NWD(A, B) + NWD(B, C) + NWD(C, A),$$

gdzie $NWD(X, Y)$ oznacza największy wspólnik dzielnik liczb naturalnych X i Y .

Rybie zależy na znalezieniu współczynników pogoni, ale niestety nie jest w stanie rozwiązać tego zadania. Twoim zadaniem jest znalezienie **dowolnej** trójki liczb, która spełnia wymagania Jerzego Ryby.

WEJŚCIE

W pierwszym i jedynym wierszu standardowego wejścia znajduje się pojedyncza liczba całkowita N oznaczająca liczbę nieudanych pogoni za Kilerem.

WYJŚCIE

W pierwszym i jedynym wierszu standardowego wyjścia powinny znaleźć się trzy liczby całkowite A, B, C spełniające warunki opisane w treści zadania. Jeśli takie liczby nie istnieją, to wypisz $A = B = C = 0$.

OGRANICZENIA

$$1 \leq N \leq 10^9.$$

PRZYKŁAD

Wejście	Wyjście
1	0 0 0
Wejście	Wyjście
10	6 10 6
Wejście	Wyjście
12	4 4 4

Czasochłonne obliczenia

Zadanie: **MMM**
Limit pamięci: **256 MB**
Limit czasu: **2 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

To by nic nie dało!

Jasio, będąc wyjątkowo znudzonym na ostatniej lekcji matematyki, zaczął wykonywać przeróżne obliczenia na ciągu, który znajdował się akurat na tablicy. Zaczął od wyliczania sumy sum elementów we wszystkich spójnych podprzedziałach tego ciągu. Szybko udało mu się jednak znaleźć wzór, który bardzo usprawnił obliczenia.

Stasio, widząc, jak mało czasu zajmują Jasiowi działania, spróbował utrudnić jego zadanie. Postanowił, że dla każdego z analizowanych podprzedziałów, Jasio może wybrać sobie dowolną nieujemną liczbę całkowitą x i zamiast bezpośrednio dodawać elementy z tablicy, obliczać sumę elementów po wykonaniu na każdym z nich operacji xor z x . Stasio chciałby, aby ostateczna suma uzyskana przez Jasia była możliwie najmniejsza. Jasio postanowił, że napisze program, który szybko obliczy wynik, o który zapytał Stasio.

Kilka dni później Stasio, napotkawszy Jasia na szkolnym korytarzu, zapytał go o postępy w obliczeniach.

– Niestety Stasiu, nie wiem jak napisać program, który szybko rozwiąże Twój problem – odpowiedział zasmucony Jasio.

– A próbowałeś może analizować przedziały od prawej do lewej, żeby szybciej liczyć wynik?

– To by nic nie dało – odpowiedział Jasio – nie zmieni to mojej złożoności asymptotycznej.

– A gdybyś analizował przedziały od najkrótszego do najdłuższego?

– Nie, to nie dało by nic – ukrócił rozmowę Jasio, po czym zmienił temat.

A czy Ty potrafisz efektywnie rozwiązać problem wymyślony przez Stasia? Musisz zwrócić jego wynik modulo $10^9 + 7$.

Formalnie, dla danego n -elementowego ciągu $\{A_i\}_{i=1}^N$ oblicz wartość

$$\left(\sum_{1 \leq l \leq p \leq N} \min_{x \in \mathbb{Z}_{\geq 0}} \sum_{i=l}^p A_i \oplus x \right) \bmod 1000000007,$$

gdzie \oplus oznacza operator xor.

WEJŚCIE

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą N .

Drugi wiersz wejścia zawiera N oddzielonych pojedynczymi spacjami liczb całkowitych A_i .

WYJŚCIE

Na wyjściu wypisz jedną liczbę całkowitą, oznaczającą wartość sumy poszukiwanej przez Stasia.

OGRANICZENIA

$1 \leq N \leq 100\,000$, $0 \leq A_i \leq 10^9$.

PRZYKŁAD

Wejście

5
1 2 3 4 7

Wyjście

67

Wejście

5

0 1 0 0 1

Wyjście

11

Przykładowo, wartością x która minimalizuje sumę dla przedziału $[1, 4]$ jest 0. Suma wynosi wtedy 1.

Dla przedziału $[2, 5]$ minimalna możliwa do uzyskania suma to 2 i jest ona osiągalna zarówno dla $x = 0$ jak i $x = 1$.

Przedział $[2, 2]$ będzie miał sumę 0 dla $x = 1$.

Test Stefana

Zadanie: **NNN**
Limit pamięci: **256 MB**
Limit czasu: **1 s**



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

Iloraz inteligencji 220 – to mało?

Stefan "Siara" Siarzewski w swojej głowie ma niezwykle wiele. Niestety, jego żona Gabrysia nie dowierza temu. Ułożyła dla swego męża specjalny test.

Przygotowała ciąg N liczb naturalnych A_1, A_2, \dots, A_N . Powiedziała mężowi, że może on wykonać co najwyżej K zamian sąsiednich elementów w tym ciągu. Stefan następnie ma podzielić ten ciąg na maksymalną liczbę spójnych przedziałów tak, aby suma liczb w każdym takim przedziale była parzysta. Gabrysia powiedziała, że jest on godny podanego ilorazu inteligencji, jeśli liczba uzyskanych przedziałów będzie największa możliwa.

Stefan ma teraz swoje kłopoty z Kilerem i nie ma czasu na testy żony. Dlatego to Twoim zadaniem jest rozwiązać to zadanie za niego.

WEJŚCIE

W pierwszej linii standardowego wejścia znajdują się dwie liczby naturalne N, K oznaczające odpowiednio długość ciągu i liczbę możliwych zamian. W drugiej i ostatniej linii standardowego wejścia znajduje się ciąg N liczb naturalnych podzielanych pojedynczym odstępem. Są to wartości kolejnych elementów ciągu A_i .

WYJŚCIE

W pierwszej i jedynej linii standardowego wyjścia Twój program powinien wypisać dokładnie jedną liczbę naturalną oznaczającą największą możliwą liczbę przedziałów możliwych do uzyskania przez Siarę. Jeśli taki podział ciągu nie jest możliwy po wykonaniu co najwyżej K zamian, należy wypisać 0.

OGRANICZENIA

$1 \leq N, K \leq 200\,000, 1 \leq A_i \leq 200\,000$.

PRZYKŁAD

Wejście

3 1
1 4 3

Wyjście

2

Przykładowy optymalny ciąg zamian wygląda następująco:

143 → 413

Uzyskujemy wówczas dwa przedziały o sumie parzystej:

[4] [13]

Wejście

5 4
5 2 4 6 3

Wyjście

4

Jeden z optymalnych ciągów zamian wygląda następująco:

52463 → 25463 → 25436 → 25346

Uzyskujemy wówczas cztery przedziały o sumie podzielnej przez 2:

[2] [5 3] [4] [6]

Wejście

4 100

3 5 4 3

Wyjście

0

Niezależnie od wykonywanego ciągu zamian, nie jesteśmy w stanie uzyskać podziału ciągu na przedziały o parzystej sumie, więc wynikiem jest 0.



Ministerstwo
Cyfryzacji



Uniwersytet
Wrocławski

To historia z życia wzięta, a zaczyna się tak...

Pewnego pięknego popołudnia do Pawła zgłosił się nie byle jaki klient. Zażądał on od niego, aby spośród dodatnich liczb całkowitych mających dokładnie N dzielników wybrał tę, która ma dla niego szczególne znaczenie. Jak wiadomo, Paweł jest fanem klubu Pierwsze Pierwszaki, dlatego postanowił on, że wybrana przez niego liczba będzie miała możliwie najwięcej różnych liczb pierwszych w rozkładzie na czynniki pierwsze.

Paweł zastanawia się, jaka może być największa liczba tych czynników. Czy jesteś w stanie mu pomóc?

WEJŚCIE

Jedyny wiersz wejścia zawiera dodatnią liczbę całkowitą N .

WYJŚCIE

Na wyjściu wypisz jedną liczbę całkowitą – największą liczbę różnych czynników pierwszych mogących wystąpić w rozkładzie dodatniej liczby całkowitej mającej dokładnie N dzielników.

OGRANICZENIA

$1 \leq N \leq 10^9$.

PRZYKŁAD

Wejście

3

Wyjście

1

Wejście

6

Wyjście

2